图书情报工作
第63卷 第4期 2019年2月

大数据环境下文本情感分析算法的规模适配研究：以 Twitter 为数据源*

■ 余传明[1]　原赛[2]　王峰[1]　安璐[3]

[1] 中南财经政法大学信息与安全工程学院　武汉 430073　[2] 中南财经政法大学统计与数学学院　武汉 430073
[3] 武汉大学信息管理学院　武汉 430072

摘要：［**目的/意义**］以大数据环境下的文本情感分析这一特定任务为目的，对规模适配问题进行研究，为情报学领域研究人员进行大数据环境下数据分析时，实现效率和成本的最优选择提供借鉴。［**方法/过程**］采用斯坦福大学 Sentiment140 数据集，在对传统情感分析算法分析的基础上，提出了 5 种面向大数据的文本情感分析算法，检验各种算法在不同环境和数据规模下的适配效果，从准确性、可扩展性和效率等方面进行实证比较研究。［**结果/结论**］实验结果显示，本文所搭建的集群具有良好的运行效率、正确性以及可扩展性，Spark 集群在处理海量文本情感分析数据时更具有效率优势，且在数据规模越大的情况下，效率优势越明显；在资源利用方面，随着节点数和核数的增加，集群的整体运行效率变化显著，配置 5 个 4 核 4G 内存的从节点，能够实现在高效完成分类任务的同时达到节约资源成本的效果。

关键词：规模适配　大数据　海量文本　情感分析　机器学习算法
分类号：TP391
DOI：10.13266/j.issn.0252-3116.2019.04.013

1 引言

随着移动互联网技术的不断发展，人们通过网络进行社交、购物等活动产生的数据量呈爆炸式增长，越来越多的用户在微博、Twitter 等社交媒体平台上发表了对娱乐、政治等事件的观点，而这些观点通常以文本为载体进行传播，如何高效地挖掘海量文本所蕴含的情感信息对企业和政府来说都是新的机遇和挑战[1]。

情感分析是一种通过机器学习自动识别用户生成内容，来判断用户对实体（如产品、人物、主题、事件等）的积极、消极或中立意见的过程。在目前研究所采用的数据集中[2-5]，评论数量规模仍然相对较小。面对海量文本，传统的模型和算法无法利用不同的计算机物理资源进行分布式并行计算，情感分析的规模适配问题成为了其所面临的瓶颈。例如，当数据规模扩展到 10 万、100 万或者 1 000 万时，情感分析方法在准确度、召回率和效率等方面是否会发生变化；情报工作人员如何将传统的情感分析算法扩展到大数据环境；情感分析算法在不同运行环境下，其运行效率具有何种差异，在实际工作中如何选择和配置最优环境。在上述背景下，规模适配问题与领域适配问题[6-7]和语言适配问题[8]相并列，成为大数据环境下观点挖掘所面临的三大严峻挑战[9]。

值得说明的是，在解决情感分析的规模适配问题上，目前相关研究仍然较多地集中在计算机科学领域[10-13]。在情报学等领域，情感分析的规模适配问题尚未引起足够的重视。鉴于此，本文在对传统情感分析算法分析的基础上，提出面向大数据的文本情感分析算法，检验各种算法在不同环境和数据规模下的适配效果，从准确性、可扩展性和效率等方面进行实证比较研究，以期为相关领域研究人员开展大数据环境下的文本情感分析，实现效率和成本的最优选择提供借鉴。

* 本文系国家自然科学基金面上项目"大数据环境下基于领域知识获取与对齐的观点检索研究"（项目编号：71373286）和教育部哲学社会科学研究重大课题攻关项目"提高反恐怖主义情报信息工作能力对策研究"（项目编号：17JZD034）研究成果之一。

作者简介：余传明（ORCID：0000-0001-7099-0853），教授；原赛（ORCID：0000-0002-5822-2496），硕士研究生；王峰（ORCID：0000-0003-1602-7235），硕士研究生；安璐（ORCID：0000-0002-5408-7135），教授，博士生导师，通讯作者，E-mail：anlu97@163.com。

收稿日期：2018-05-09　修回日期：2018-09-21　本文起止页码：101-111　本文责任编辑：杜杏叶

2　研究现状

情感分析在不同数据规模上的研究重点不同。在小规模数据集上,情感分析的研究重点是提升分析粒度,尽可能在更细粒度上进行情感分析,并将其扩展到更多的应用场景中。在大规模数据集上,通常直接采用统计学习的方式来解决情感分析问题,其重点在于机器学习算法的改进及其并行化实现。

在小规模数据集上,文本情感分析的实际应用包括面向情感词汇的应用、面向机器学习的应用和面向扩展的应用等。在面向情感词汇的应用方面,主要通过抽取文本中的情感词汇及其对应的评价实体等,并在此基础上进行情感极性判断,包括情感词典方法、本体方法和信息抽取方法等。情感词典方法通过情感词典判断文本中词的情感倾向从而对文本倾向进行分析[14-15];本体方法通过形式化定义概念及概念之间的关系,以提高模型的识别效果[16-17];信息抽取方法则是使用自动化方法抽取情感词,从而进行情感分析[18-19]。这些研究通常粒度较细,且需要较多的人工参与,因而适用于规模较小的数据。面向机器学习的应用,则跳过较为繁琐的情感词汇构建过程,而直接采用统计学习方式来解决情感分析问题[6,8,20-24]。由于避免了较为繁琐的情感词汇构建过程,因而更容易适配到规模较大的数据集。在扩展应用方面,目前情感分析在客户评论挖掘[25]、网络舆情管理[26-27]、顾客满意度调查[28]、网络谣言识别[29]和竞争对手识别[30]等情报领域得到了广泛应用,对具体应用场景的依赖程度高于数据规模和算法。

在大规模数据集上,目前研究一是侧重情感分析算法理论的纵向创新,即结合大数据平台(例如 Spark 计算框架)对算法进行改进,在提高分类效率的同时保持或提升分类准确度。例如,朱继召等[31]在 Spark 框架中设计分布式 CRFs 算法 – SparkCRF 用于文本分析,结果表明 SparkCRF 具有良好的计算能力和扩展性能,并能达到与传统单节点同水平的准确率。J. Chen 等[32]设计了 Spark 平台上的并行随机森林(PRF)算法,通过在训练过程中采用降维方法以及预测过程中采用加权投票方法,提高了算法对大型、高维和噪声数据的分类准确性,结果表明 PRF 算法在分类准确性和可扩展性方面优于 Spark MLlib 中的分类算法。二是侧重于在大数据平台上对各种分类算法进行横向比较,以寻求不同资源配置下特定文本数据的最优分类算法,此时分类效果受数据属性、数据集规模及 Spark 资源配置三方面影响,其评估方式以比较不同分类算法的准确度和加速比为主。例如,G. Mogha 等[13]比较了单机 Spark 中决策树、朴素贝叶斯、随机森林和支持向量机算法在航空公司数据集上分类准确性,结果表明决策树分类效果要优于其他三种分类算法。A. Baltas 等[1]在 Spark 平台中采用 MLlib 对 Twitter 数据进行情感分析,对于二元和三元情感数据,朴素贝叶斯的分类效果优于逻辑回归和决策树,同时数据集大小对朴素贝叶斯分类效果影响显著。M. Hai 等[33]研究在集群 Spark 中朴素贝叶斯和随机森林两种分类算法的不同表现,并给出不同数据集对应的最佳 Spark 节点数量,结果表明两种分类算法的准确率都很高,且对于不同大小的数据集,加速比最大时节点数量不同。

在上述背景下,本文以 Spark 平台作为基础,尝试研究不同情感分析算法在不同规模数据集、不同集群配置条件下的计算效率、准确性和可扩展性,以期为相关领域的大数据分析人员合理选择算法和配置平台提供建议,实现效率和成本的最优选择。

3　研究问题与研究方法

3.1　研究问题

本文研究大数据环境下文本情感分析算法的规模适配问题,即在领域和语言相同的情况下,通过对传统情感分析算法进行分析,提出面向大数据的文本情感分析算法,检验在不同环境和不同数据规模下的适配效果,从计算效率、准确性和可扩展性等方面进行比较分析。具体而言,本文提出以下研究问题:

(1)情感分析算法在不同运行环境下(传统 Sklearn 方式、Spark 单节点方式和 Spark 集群方式),其运行效率具有何种差异;相比 Spark 单节点方式和传统 Sklearn 方式,Spark 集群方式在处理海量情感分类数据时是否更具有优势,且是否数据规模越大,优势越明显;在构建大数据情感分析平台时,如何根据上述差异来更好地配置运行环境。

(2)各种情感分析算法在不同运行环境下,其运行效果(以正确率、召回率、F 值等来衡量)具有何种差异;相比 Spark 单节点方式和传统 Sklearn 方式,Spark 集群方式在处理海量情感分类数据时是否具有更好的效果。

(3)随着节点数和核数的增加,集群的整体运行效率如何变化;在配置大数据情感分析平台时,如何根据上述变化来更好地设置最佳的核数和节点数,以通过较小的开销来获得较高的效率。

（4）当数据集在小于何种规模时，并行式算法的运行加速比小于 1，即采用并行式算法采用集群处理不能发挥其计算优势；换言之，当情感分析的数据规模达到何种程度时，有必要采取并行式情感分析算法。

本文之所以强调文本情感分析，主要基于以下原因：①在不同数据规模上，情感分析的研究重点有所不同。针对小规模数据集，重点在于提升分析粒度并将其扩展到更多的应用场景；针对大规模数据集，重点在于机器学习算法改进及其并行化实现。本文尝试对情感分类算法进行改进，并提出其并行化实现，落脚点在于情感分析。②尽管文本情感分析中的情绪分类或者正负倾向性分类都可以归结为文本分类问题，但本文以 Twitter 作为数据源，主要任务仍然是情感分析。

3.2 大数据环境下的文本情感分析平台构建

目前，分布式计算平台包括 Spark、Hadoop、GridGain、Mars、Phonenix、Twister、Disco、HaLoop、iMapReduce、iHadoop、PrIter 和 Dryad 等[34]。相比于其他平台，Spark 并行平台采用基于内存的迭代处理操作，其计算效率、容错性和可扩展性更好；Spark MLlib 库中提供了常用的机器学习算法以及 Python 编程环境，便于比较传统 Sklearn 和并行 Spark 之间的差异。鉴于此，本文采用 Spark 体系结构，并在此基础上开展实验。

本文所采用的大数据文本情感分析框架见图 1，系统框架包括驱动程序（Driver Program），集群管理器（Cluster Manager）和工作节点（Worker Node）三个组件[35]。驱动程序是一个定义上下文（SparkContext）的客户端程序，负责向群集管理器请求资源，并将相应的包文件（jar 文件）以及配置的依赖项提交给工作节点。集群管理负责分配资源并跟踪提交作业的状态，在独立运行模式（Standalone 模式）中作为主节点（Master Node）控制整个集群，监控工作节点，作为资源管理组件。工作节点又称从节点，用于负责控制计算节点，包括任务（Task）和缓存（Cache），通过启动执行器（Executor），计算并返回相应结果。

具体而言，本文相关的实验在青云服务器平台进行，分别配置了单机和集群环境，具体信息如表 1 所示。单机服务器配置 4 核 CPU 和 4G 内存，并用 Python3.6 执行传统 Sklearn 方式和 Spark 单节点运行程序；集群服务器由四种节点构成，Client 节点配置 2 核 CPU 和 2G 内存，用于存储本地数据及提供程序运行界面，其计算性能不影响集群效率；HDFS 节点同样配置 2 核 CPU 和 2G 内存，用于存储 Client 节点的用户上

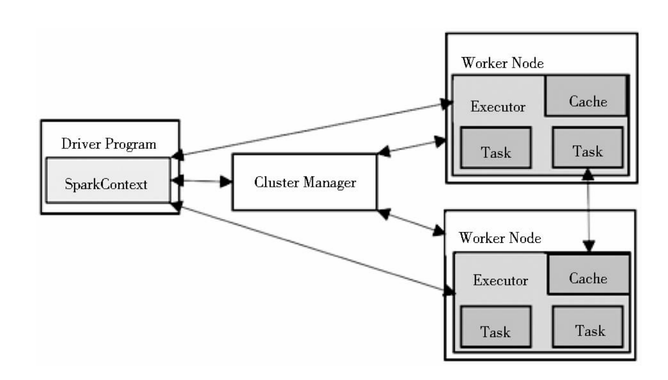


图 1 Spark 文本情感分析框架[35]

传数据并为集群提供底层数据存储，其计算性能同样不影响集群效率；主节点配置 4 核 CPU 和 4G 内存，在 Standalone 模式中作为资源管理组件控制整个集群，为便于不同运行环境比较，采用与单机相同配置；从节点作为集群计算节点，其不同的资源配置决定了情感分析算法的效率，从节点个数变化范围 3－7 个，并根据实验具体需求配置内存为 4G，CPU 核数为 2、4 或 8 个。

表 1 服务器平台配置信息

环境	角色	个数	系统	内存/G	核数/个	系统盘/G	软件环境
单机	单节点	1	Ubuntu Server 14	4	4	100	Spark 2.2.0 Hadoop 2.7.3 Python3.6
集群	Client 节点	1		2	2	10	
	HDFS 主节点	1		2	2	10	
	主节点	1		4	4	10	
	从节点	7		4	2/4/8	10	

3.3 大数据环境的文本情感分析算法

本文在决策树（DT）、逻辑斯蒂回归（LR）、朴素贝叶斯（NB）、随机森林（RF）和支持向量机（SVM）的基础上，提出并行决策树算法、并行逻辑回归算法、并行朴素贝叶斯算法、并行随机森林算法和并行支持向量机算法。之所以选择 DT 等算法作为研究基础，主要基于以下原则：①可用性，即算法在传统的文本情感分析中得到广泛应用；②可扩展性，即算法能够扩展到大规模数据环境；③可比性，即算法在准确性和效率等方面具有可比性。值得说明的是，除了本文所采用的五种分类算法，还存在其他的并行分类算法，例如梯度提升树、多层感知机等。由于梯度提升树中多个弱学习器之间存在依赖关系，会导致并行训练效果不佳；而多层感知机由于隐藏层的非凸损失函数可能存在多个局部最小值，不同随机权重初始化可能导致不同的实验结果，此外多个超参数的调整不利于不同并行算法的比较，因此在本文实验中未加以采用。

3.3.1 并行决策树算法 并行决策树采用分治策略，通过反复迭代将原有问题分为多个特征相似的子问题求解，并行决策树生成过程包括三种机制，即最优属性计算、树节点并行、数据集并行[36]。由于各节点属性独立，使得当前节点各属性信息增益可有不同处理器计算，然后由主处理器统一决策出最优属性；或者可以将分配到不同处理器的各属性数据综合处理，得到最优属性。树节点并行指依据最优属性采用并行化操作将树节点划分成若干子节点，相互独立的不同子节点在下一步划分中可以并行进行。对于不同处理器中处理的数据子集采用同一决策树算法处理，将分类模型保存并提供给其他处理器后续使用。在文献[36]的基础上，本文提出的并行决策树算法如算法1所示：

算法1 基于Spark的并行决策树算法

输入：数据集T

1 从HDFS上读取数据集并转化为RDD

2 使用map算子获得情感得分和文本向量

3 使用Tokenizer算子将文本向量转化为词向量

4 计算词向量的TF-IDF，其中特征数设为3 000

5 获取标签列和特征列并进行索引，得到预处理后的数据集

6 将预处理后数据集随机划分，其中80%用于训练、20%用于测试，并设随机数为0

7 使用最大树深为10的决策树分类器对训练数据拟合，并记录训练时间

8 使用训练模型对测试数据预测分类，并输出精度、准确率、召回率及F1值

输出：情感分析结果

3.3.2 并行逻辑回归算法 并行逻辑回归算法主要基于Bagging抽样实现并行，并采用有限内存LBFGS对估计系数进行加速优化[12]。Bagging算法可以生成具有相同权重的多个分类器，并通过多次投票将它们聚合成一个预测器，预测器通过整合来自多个逻辑回归分类器的预测结果来计算一个实例的最终预测。在文献[12]的基础上，本文提出的并行逻辑回归算法如算法2所示：

算法2 基于Spark的并行逻辑回归算法

输入：数据集T

1 从HDFS上读取数据集并转化为RDD

2 使用map算子获得情感得分和文本向量

3 计算文本向量TF-IDF值

4 使用zip算子将TF-IDF向量和情感得分连接并转化为LabeledPoint类型

5 将转换后的数据集随机划分，其中80%用于训练、20%用于测试，并设随机数为0

6 使用最大迭代步长为1 000的逻辑回归分类器对训练数据拟合，并记录训练时间

7 使用训练模型对测试数据预测分类，并输出精度、准确率、召回率及F1值

输出：情感分析结果

3.3.3 并行朴素贝叶斯算法 并行朴素贝叶斯算法通过将训练集和测试集分配给多个节点进行模型构建与分类预测[37]。首先计算训练集中特征项在各类中出现的TF以及每个特征项对应的TF-IDF值，然后计算每个类别的先验概率及TF-IDF总和，以及每个类别下特征项的条件概率，生成并行朴素贝叶斯模型，最后根据测试集和输入模型计算每条测试文本所属类别的条件概率，得到最终分类结果。在文献[37]的基础上，本文提出的并行朴素贝叶斯算法如算法3所示：

算法3 基于Spark的并行朴素贝叶斯算法

输入：数据集T

1 从HDFS上读取数据集并转化为RDD

2 使用map算子获得情感得分和文本向量

3 使用Tokenizer算子将文本向量转化为词向量

4 计算词向量的TF-IDF，其中特征数设为3 000

5 获取标签列和特征列并进行索引，得到预处理后的数据集

6 将预处理后数据集随机划分，其中80%用于训练、20%用于测试，并设随机数为0

7 使用平滑度为1、模型类别为multinomial的朴素贝叶斯分类器对训练数据拟合，并记录训练时间

8 使用训练模型对测试数据预测分类，并输出精度、准确率、召回率及F1值

输出：情感分析结果

3.3.4 并行随机森林算法 并行随机森林算法分两个阶段完成[38]。第一阶段训练随机森林分类器，在Spark中将向量化训练集转化为RDD分发给各节点，各节点将对其采用Bagging抽样，抽样次数由决策树棵树K决定；随后对抽样样本建立决策树，通过union算子将分散的决策树汇总生成随机森林。第二阶段对测试集进行分类，各节点中所有决策树对测试集中每条样本进行投票，选择判定类别次数最多的类作为最终分类。最后将分类结果保存在HDFS上。在文献[38]的基础上，本文提出的并行随机森林算法如算法4所示：

算法 4 基于 Spark 的并行随机森林算法

输入:数据集 T

1 从 HDFS 上读取数据集并转化为 RDD

2 使用 map 算子获得情感得分和文本向量

3 使用 Tokenizer 算子将文本向量转化为词向量

4 计算词向量的 TF-IDF,其中特征数设为 3 000

5 获取标签列和特征列并进行索引,得到预处理后的数据集

6 将预处理后数据集随机划分,其中 80% 用于训练、20% 用于测试,并设随机数为 0

7 使用树的棵树为 10、最大树深为 10 的随机森林分类器对训练数据拟合,并记录训练时间

8 使用训练模型对测试数据预测分类,并输出精度、准确率、召回率及 F1 值

输出:情感分析结果

3.3.5 并行支持向量机算法 并行支持向量机训练流程如下。在文献[39]的基础上,首先使用 RDD 中的 textFile 函数读取 HDFS 中的数据并转换为 RDD 数据类型,并将其随机切分为合适大小的数据块。Map 操作将格式化的训练数据并行 SVM 训练,repartition 函数对训练结果进行整合并划分本次输入的数据块,不断迭代进行 SVM 训练并得到满足条件的全局最优支持向量分类器,最后根据生成的分类器对测试数据进行分类。本文提出的并行 SVM 算法如算法 5 所示:

算法 5 基于 Spark 的并行 SVM 算法

输入:数据集 T

1 从 HDFS 上读取数据集并转化为 RDD

2 使用 map 算子获得情感得分和文本向量

3 计算文本向量 TF-IDF 值

4 使用 zip 算子将 TF-IDF 向量和情感得分连接并转化为 LabeledPoint 类型

5 将转换后的数据集随机划分,其中 80% 用于训练、20% 用于测试,并设随机数为 0

6 使用最大迭代步长为 1 000 的支持向量机分类器对训练数据拟合,并记录训练时间

7 使用训练模型对测试数据预测分类,并输出精度、准确率、召回率及 F1 值

输出:情感分析结果

3.4 参数设置

实验预处理阶段,选用 80% 数据用于训练、20% 数据用于预测,且同一实验重复运行 3 次,以生成训练模型所需时间平均值为运行时间。在参数设置方面,遵循以下原则,一是可用性,即参数选择使得模型可用,并具有相对较好的分类效果;二是对等性,即在保证算法分类效果的前提下,使算法在不同环境(包括传统 Sklearn 环境、单节点和多节点环境等)下的参数设置尽可能相近,以便于对并行算法分类效率进行有效比较。DT 算法特征数设为 3 000,最大树深为 10;LR 算法迭代步长为 1 000;NB 算法特征数为 3 000,模型类型为“multinomial”;RF 算法特征数为 3 000,树棵数和最大树深为 10;SVM 算法迭代步长为 1 000。

3.5 评价指标

情感分析算法的评价指标由效果和效率两方面构成。分类效果采用精度、准确率、召回率和 F1 值等指标。分类效率包括运行时间和加速比等指标。

本文中,运行时间(Time)衡量的是分类算法生成训练模型所需时间,加速比(Speedup)是衡量在单节点模式下和并行模式下运行同一任务所耗费的时间的比率,通过改变节点数量并保持数据集不变来计算加速比。T_1 代表在单节点上分类算法的运行时间,T_m 代表相同数据集条件下在 m 个节点上相同算法的运行时间,如果加速比随着节点数量的增加而线性增加,则意味着多个节点可以有效缩短算法的运行时间。

$$Speedup = T_1/T_m \qquad 式(1)$$

4 实验结果与讨论

4.1 实验数据

海量文本情感分类一直是微博领域富有挑战性的话题,由于 Twitter 发文具有短篇幅的特点,更容易表达 Twitter 用户的情感状态。本文选取斯坦福大学 Twitter 推文情感数据集[40-41]用于情感分析,主要基于以下考虑。

首先,该数据集在文本情感挖掘、社交网络分析中得到较多的应用。例如 B. Heredia 等[42]使用该推文数据集及评论数据集进行跨领域情感挖掘;A. Goel 等[43]在研究推特实时情绪分析中使用该数据集训练朴素贝叶斯模型并提出分类准确性更好的算法;M. L. Lima 等[44]使用该数据集作为语料库,通过推特情绪分析进行股票交易预测;N. Friedrich 等[45]从该数据集中抽取出具有代表性的学术推文,通过分析情绪得出 Twitter 有助于提高学术交流与传播的结论。

其次,该数据集能够划分为不同的规模。由于本文研究的重心在于比较不同环境下不同算法在海量短文本情感分类中的差异,考虑到数据集选取的有效性前提下,能够将数据集划分为不同规模,以展开情感分析的规模适配研究。

该数据集包含 1 578 627 条 Twitter 推文,每条推文用 1 表示积极情绪,0 表示消极情绪。从原始数据集中分别提取出 15M、30M、75M、150M、300M 大小的数据(对应的 Twitter 条数分别为 16 万、31 万、78 万、158 万和 316 万),用于研究不同数据规模在集群环境下的情感分析算法间的差异。

4.2 实验结果

为了探究大数据环境下情感分类的规模适配问题,我们评估 5 种传统情感分析算法(DT、LR、NB、RF、SVM)及相应的 5 种并行情感分析算法,在传统 Sklearn、单节点 Spark 以及集群 Spark 环境下运行效率、正确性和可扩展性三方面表现。

4.2.1 运行效率 实验一比较 5 种分类算法在传统、单节点 Spark 和集群 Spark 运行效率。这里使用 150M 中等数据集进行实验,传统和单节点 Spark 均采用 4 核 4G 运行环境,集群 Spark 采用 3 个 4 核 4G 的从节点,且实验数据由 HDFS 主节点读取并存储。图 2 展示了不同运行环境的分类算法运行时间差异:

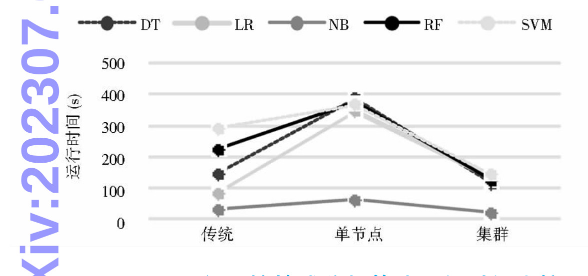


图 2 不同运行环境情感分析算法运行时间比较

由图 2 可以看出,不同环境下分类算法运行效率由低到高依次为:Spark 单节点方式、传统 Sklearn 方式和 Spark 集群方式。Spark 单节点方式比传统 Sklearn 方式运行时间更长,主要是因为单节点 Spark 采用 Pyspark 模块,将本地存储数据转换为 RDD 形式时由于资源限制会耗费大量时间,而传统 Sklearn 则通过 numpy 模块直接将本地数据存储以 array 形式进行运算,故而会比单节点 Spark 更有效率。集群 Spark 比传统 Sklearn 运行时间更短,耗时缩短将近一半,主要是因为在集群模式下,Spark 引入弹性分布式数据集 RDD 将中间数据存储到从节点内存中,由于不同操作可以直接从内存中读取操作结果,因此相较于传统 Sklearn 可以明显提升机器学习算法迭代性能和数据分类效率。

4.2.2 正确性 实验二比较 5 种分类算法在传统、单节点 Spark 和集群 Spark 中的正确性。实验数据采用 150M 数据集,正确性评估的指标包括精度、准确率、召回率和综合指标 F1 值。图 3 展示了不同运行环境分类算法 F1 值用于比较正确性差异:

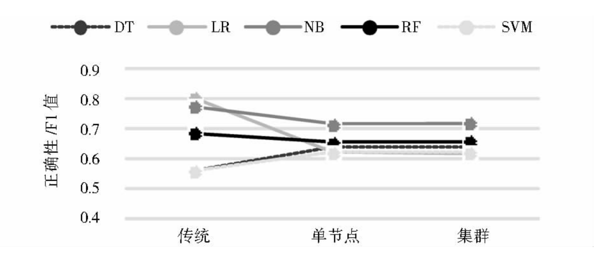


图 3 不同运行环境情感分析算法正确性比较

由图 3 可以看出,除 DT、SVM 算法外,传统 Sklearn 方式各项正确性指标略优于 Spark 单节点方式和 Spark 集群方式。以 LR 算法的 F1 值为例,传统 Sklearn 达到 0.801 9,而单机和集群分别为 0.618 1、0.617 2,这主要是由于 Sklearn 和 Pyspark 对预测数据处理策略不同造成的。此外,Spark 单节点方式和 Spark 集群方式正确性非常接近,说明对于 Spark 而言,从单节点模式扩展到集群模式并不会降低分类准确性。在各种分类算法中,分类正确性效果最好的是 NB 算法,其 F1 值分别达到 0.715 3、0.717 3,相比于传统 Sklearn 的 SVM 算法,其 F1 值提升了 3% 左右。

4.2.3 可扩展性 实验三比较集群 Spark 中运行时间随从节点核数变化情况,以评估集群的可扩展性。实验数据采用 150M 数据集,同时在 3 个从节点条件下,以处理器核数为可变因素,分别设为 2 核、4 核、8 核。这里图 4 展示了算法运行时间随从节点核数变化情况:

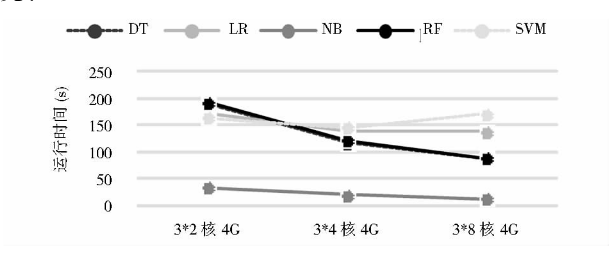


图 4 情感分析算法运行时间随从处理器核数变化情况图

由图 4 可以看出,除 SVM 算法运行时间先减少后增加外,其余四种分类算法的运行时间随核数增加而减少。实验总体符合预期假设,即:随着处理器核数增加,集群拥有更多资源用于任务执行,必然会提高整体运行效率,表现出良好的可扩展性。此外,不同分类算法运行时间具有一定的差距性,同时随着处理器核数的增加,运行时间下降速率非线性变化,因此集群环境中 4 核比 8 核对计算的资源利用率更高。从各种算法

的时间对比来看,实验中表现最好的是 NB 算法,其运行时间分别为 33.14s、19.59s、12.12s,少于其他四种算法。

实验四比较集群 Spark 中运行时间随从节点个数变化情况,从另一方面评估集群的可扩展性。同样采用 150M 数据集,实验设置每个从节点为 4 核,以从节点个数为可变因素,分别设为 3 个、4 个、5 个、6 个、7 个。图 5 展示了算法运行时间随从节点个数变化情况:

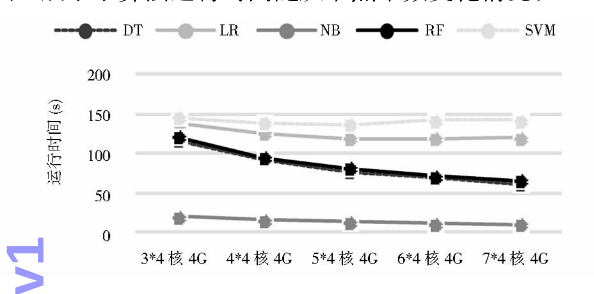


图 5 情感分析算法运行时间随从节点个数变化情况

由图 5 可以看出,当从节点个数增加到 5 个以后,SVM 和 LR 算法的运行时间略有增加,而 DT、NB 和 RF 算法的运行时间减少趋势放缓,这表明主节点与多个节点数量间的通信时间是制约集群可扩展性的因素之一。相比于实验三,随着从节点个数增加,运行时间的变化趋势减缓,说明从节点个数在不同量级的改变对实验有一定的影响。从资源利用率角度来看,各分类算法在 5 个从节点条件下均表现出良好的运行效率。从各种算法的时间对比来看,表现最好的是 NB 算法,其运行时间分别为 19.59s、14.92s、12.52s、11.31s、9.84s,少于其他四种算法。

实验五比较单节点和集群 Spark 中运行时间随数据规模变化情况,以评估集群的可扩展性。实验数据分别采用 15M、30M、75M、150M 和 300M 数据集,单节点 Spark 采用 4 核 4G 环境,集群 Spark 每个从节点配置 4 核 4G 环境。表 2 给出了不同数据规模下单节点、多种从节点运行时间比较结果,图 6、图 7 分别展示了单节点、5 个从节点不同算法的运行时间情况。

从表 2 可以看出,随着数据规模的增加,单节点运行时间近似成比例增加,多节点中除 SVM 算法外,其余算法的运行时间也呈现按比例增加的现象。从单节点运行表现来看,NB 算法运行时间最短,DT、LR 和 RF 三种算法的运行时间变化趋势较为相似,而 SVM 算法随用的运行时间最长。从多节点运行表现来看,当数据规模在 15M 和 30M 时,采用不同节点的同种分类算法运行时间没有显著差异,这表明 Spark 集群在小数据集上并没有完全利用计算资源。同时可以看出,随

表 2 不同数据规模下单节点、多种从节点运行时间比较(时间:s)

环境	算法	数据规模				
		15M	30M	75M	150M	300M
单节点	DT	59.909 6	97.851 4	200.974 0	386.861 3	730.551 3
	LR	53.793 7	62.340 5	116.034 7	345.057 8	654.236 6
	NB	7.835 3	14.376 9	33.886 2	62.683 9	125.865 8
	RF	61.452 4	98.710 7	207.915 6	374.263 9	731.158 1
	SVM	73.567 6	71.720 6	144.051 0	366.063 0	793.221 1
3 个从节点	DT	41.222 6	41.065 8	68.823 7	116.081 1	216.361 9
	LR	56.783 6	66.001 2	100.358 1	137.453 9	188.997 0
	NB	5.588 1	6.457 7	11.231 0	19.592 5	36.011 5
	RF	42.917 4	42.917 7	69.739 0	121.137 8	218.445 2
	SVM	79.415 9	87.335 5	155.726 3	144.669 4	278.027 6
4 个从节点	DT	38.420 5	37.663 4	53.723 7	91.600 6	164.631 9
	LR	55.757 2	64.094 0	97.808 4	125.644 4	170.280 5
	NB	5.414 5	6.008 5	9.020 4	14.923 6	27.538 0
	RF	39.442 7	39.473 7	54.958 0	93.860 3	176.519 0
	SVM	76.645 1	84.952 7	155.609 7	137.954 6	165.476 3
5 个从节点	DT	38.714 0	38.768 0	47.089 9	76.084 6	139.478 1
	LR	55.315 6	63.845 4	92.895 5	118.769 7	155.102 7
	NB	6.642 4	5.904 8	7.746 1	12.529 0	22.975 2
	RF	40.212 6	40.660 7	50.959 6	80.432 8	140.480 4
	SVM	76.966 3	85.632 7	154.996 4	136.043 0	157.388 8
6 个从节点	DT	40.754 6	38.354 3	48.830 4	69.157 2	117.309 9
	LR	56.014 5	63.911 7	90.793 8	118.308 8	146.847 2
	NB	5.849 6	5.723 9	7.148 6	11.316 4	20.673 7
	RF	42.293 2	41.966 6	45.927 6	70.472 8	121.909 3
	SVM	77.289 0	86.885 8	157.197 1	142.184 7	159.828 0
7 个从节点	DT	44.436 8	40.503 0	42.370 6	60.394 3	103.761 5
	LR	57.432 1	64.323 5	91.475 4	119.147 5	141.436 7
	NB	6.577 3	5.653 2	6.769 7	9.843 3	17.898 2
	RF	41.896 1	41.272 3	44.757 6	65.639 9	108.283 8
	SVM	80.416 9	87.548 9	156.424 6	142.286 6	163.209 6

着从节点数量的增加,同种算法的运行时间均有所减少,而增加到 5 个以后,运行时间减少的趋势变慢,这表明多节点运行效率受节点间通信时间影响。不同节点实验中,NB 算法运行效率最高。

从图 6 可以看出,单节点情感分析算法运行时间随数据规模变大而增加。当数据规模小于 75M 时,各种分类算法运行时间增加较为缓慢;当数据规模增加到 75M 之后,各种分类算法运行时间迅速增加。这表明在处理大规模数据集分类时,单节点 Spark 并不能有效发挥其并行计算优势。此外,随着数据规模的增加,不同分类算法间的运行差异进一步加大,SVM 算法运行增加时间最多,而 NB 算法运行时间增势较为平缓,且运行时间最短。这表明随着数据规模的增加,

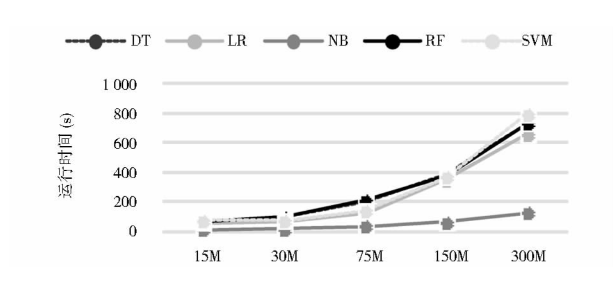


图 6 单节点情感分析算法运行时间随数据规模变化情况

情感分析的算法的选择与改进变得更加重要。

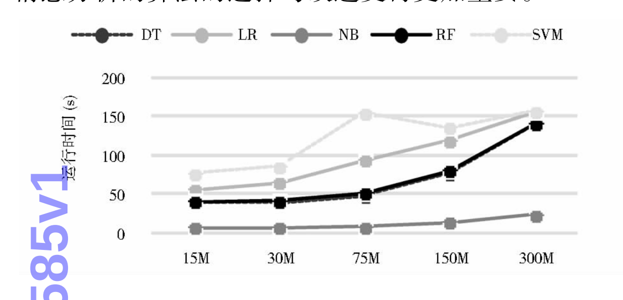


图 7 多节点情感分析算法运行时间随数据规模变化情况

由图 7 可以看出,在五个多节点环境下,随着数据集规模增大,各种算法运行时间均有所增加,特别是当数据集增加到 75M 以后,运行时间增加更为明显,这与实验预期假设相符合,即:数据规模是影响情感分类算法运行效率的因素之一,且数据规模越大,分类时间越长。在数据规模增大过程中,SVM 算法运行时间具有不稳定性,同时在所有分类算法中运行效率最低。相比之下,各种数据规模性下分类效率最高的均为 NB 算法,其运行时间分别为 6.64s、5.90s、7.74s、12.52s、22.97s,少于其他算法。

由图 8 可以看出,随着数据集规模增大,各种算法运行加速比明显增加,这表明 Spark 集群方式在处理海量情感分类数据时比 Spark 单节点方式更具有优势,且数据规模越大,加速效果越明显。同时可以看到,当数据集小于 75M 时,LR 和 SVM 算法的运行加速比小于 1,说明在小数据上这两种算法采用集群处理并没有发挥其计算优势。

4.3 讨论

本文在决策树(DT)、逻辑斯蒂回归(LR)、朴素贝叶斯(NB)、随机森林(RF)和支持向量机(SVM)的基础上,提出了并行决策树算法、并行逻辑回归算法、并行朴素贝叶斯算法、并行随机深林算法和并行支持向量机算法,并将其应用到大数据环境下的情感分析实证研究中。在上述实证研究中,以下问题值得讨论与

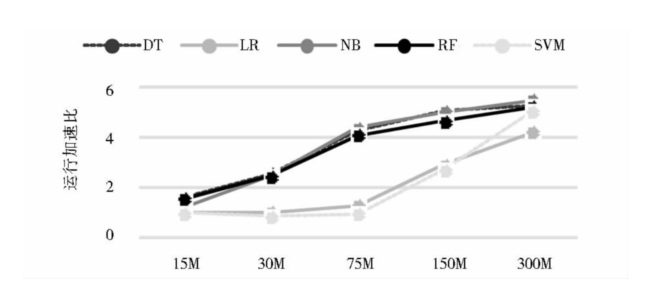


图 8 情感分析算法运行加速比随数据规模变化情况图

关注。

首先,各种情感分析算法在不同的运行环境下(传统 Sklearn 方式、Spark 单节点方式和 Spark 集群方式),其运行效率具有明显差异。相比 Spark 单节点方式和传统 Sklearn 方式,Spark 集群方式在处理海量情感分类数据时更具有优势,且在数据规模越大的情况下,优势越明显。这表明,在大数据环境下展开情感分析研究,有必要采取 Spark 集群方式,以节省时间和提升效率。

其次,各种情感分析算法在不同运行环境下,其运行效果(包括正确率、召回率、F 值等)具有显著差异。相比 Spark 单节点方式和传统 Sklearn 方式,Spark 集群方式在处理海量情感分类数据时并不能显著提升情感分析的准确率、召回率和 F 值等指标。相反地,对于多数情感分析算法而言,当从传统 Sklearn 方式转换到 Spark 单节点方式或者集群方式时,其各项指标均有所下降。这表明,在大数据环境下展开情感分析研究,如果采取 Spark 集群方式,在准确率、召回率和 F 值等情感分析效果指标提升上仍然需要做大量工作。

再次,在集群环境下,随着节点数和核数的增加,集群的整体运行效率呈现变化。一方面,随着节点数和处理器核数的增加,集群拥有更多资源用于任务执行,能够显著提高整体运行效率,表现出良好的可扩展性。另一方面,随着节点数和处理器核数的增加,运行时间下降速率呈现非线性变化;换言之,在节点数(处理器核数)达到一定阈值以后,情感分析的效率提升有所减缓。在实证研究中,上述阈值的选择取决于数据量的大小。在本文对应的数据量情况下,当核数设为 4 和节点数设为 5 时,能够通过较小的开销来获得较高的效率。实证研究表明,在大数据环境下展开情感分析研究时,并非节点数和处理器核数越多越好;相反的,采取与特定数据规模的阈值最为接近的节点数和处理器核数,能够在最大限度节省系统资源的情况完成情感分析任务。

最后，从数据量大小来看，当数据集在小于一定规模时，并行式算法的运行加速比有可能小于 1，表明采用并行式算法采用集群处理不能发挥其计算优势；换言之，当情感分析的数据规模达到一定程度时，有必要采取并行式情感分析算法。在本文的实证研究中，当数据集小于 75M 时，多数情感分析算法的运行加速比小于 1，表明采用集群处理并没有发挥其计算优势。上述实证研究表明，在情感分析研究中，有必要对数据量的大小进行界定，当数据量低于一定阈值时，采取 Spark 集群方式，并不能发挥其计算优势；相反的，反而会浪费大量的系统资源。在这种情况下，建议采用传统的 Sklearn 方式。

总的来说，本文的规模适配研究基于文本情感分析这一具体任务而提出，但实证研究所得出的结论并非局限在文本情感分析这一领域，可推广到与大规模文本信息处理相关的领域，包括文本分类、产品推荐、信息检索和数据挖掘等。例如，在大数据环境下展开文本分类或产品推荐研究时，可以采取与特定数据规模的阈值最为接近的节点数和处理器核数，能够在最大限度节省系统资源的情况完成任务；在数据量大于一定规模时，则有必要慎重选择并行式分类或推荐算法，以避免因算法选择失误而导致较多的时间消耗。值得说明的是，本文所采用的 Spark 虽然具有基于内存进行迭代计算的优点，但其数据分区能力有限，各台机器计算任务分配不均时仍然会导致负载失衡，因此并非适用于所有的数据规模、数据类型和算法。本文在 300 兆文本类型数据上对五种算法进行了验证，在后续工作中，我们将对更大的数据规模、更多的数据类型和算法进行检验，以进一步探讨规模适配下并行任务分配和算法应用的效果。

5 结语

本文以 Twitter 数据为例，在对传统情感分析算法进行分析的基础上，提出了 5 种面向大数据的文本情感分析算法，检验各种算法在不同环境和数据规模下的效果，从准确性、可扩展性和效率等方面进行实证比较研究。实验结果表明，本文所搭建的集群具有良好的运行效率、正确性以及可扩展性，Spark 集群在处理海量文本情感分析数据时更具有效率优势，且在数据规模越大的情况下，效率优势越明显；在资源利用方面，随着节点数和核数的增加，集群的整体运行效率变化显著。实验结果表明，配置 5 个 4 核 4G 内存的从节点，能够实现在高效完成分类任务的同时达到节约资源成本的效果。

本文的不足之处在于：①实验数据集较为单一，在后续工作中，我们将采用更多样的数据集对情感分析的规模适配问题进行更为深入的研究；②在资源配置方面，对子节点核数、个数进行不同量级扩充，以进一步探究大规模集群环境下不同情感分析算法的最优资源配置策略；③对领域与语言变化情境下的规模适配问题进行扩展研究。

参考文献：

[1] BALTAS A, KANAVOS A, TSAKALIDIS A K. An Apache Spark implementation for sentiment analysis on Twitter data[C]// Proceedings of algorithmic aspects of cloud computing. Cham: Springer, 2016:15－25.

[2] 明均仁. 融合语义关联挖掘的文本情感分析算法研究[J]. 图书情报工作,2012,56(15):99－103.

[3] 唐晓波,兰玉婷. 基于特征本体的微博产品评论情感分析[J]. 图书情报工作,2016,60(16):121－127.

[4] XU H, ZHANG F, WANG W. Implicit feature identification in Chinese reviews using explicit topic mining model[J]. Knowledge-based systems, 2015,76(3):166－175.

[5] 刘雯,高峰,洪凌子. 基于情感分析的灾害网络舆情研究——以雅安地震为例[J]. 图书情报工作,2013,57(20):104－110.

[6] 余传明. 基于深度循环神经网络的跨领域文本情感分析[J]. 图书情报工作,2018,62(11):23－34.

[7] 余传明,冯博琳,安璐. 基于深度表示学习的跨领域情感分析[J]. 数据分析与知识发现,2017(7):73－81.

[8] 余传明,冯博琳,田鑫,等. 基于深度表示学习的多语言文本情感分析[J],山东大学学报：理学版,2018,53(3):13－23.

[9] 余传明, 安璐. 从小数据到大数据——观点检索面临的三个挑战[J]. 情报理论与实践, 2016, 39(2):13－19.

[10] 向小军, 高阳, 商琳,等. 基于 Hadoop 平台的海量文本分类的并行化[J]. 计算机科学, 2011, 38(10):184－188.

[11] GLUSHKOVA D, JOVANOVIC P, ABELLO A. MapReduce performance model for Hadoop 2.x[EB/OL]. [2017－12－30]. https://doi.org/10.1016/j.is.2017.11.006.

[12] PAN J, HUA Y, LIU X, et al. Bagging-based logistic regression with Spark: a medical data mining method[C]// International conference on advances in mechanical engineering and industrial informatics. Atlantis: Atlantis Press,2016:1553－1559.

[13] MOGHA G, AHLAWAT K, SINGH A P. Performance analysis of machine learning techniques on big data using Apache Spark[C]// International conference on recent developments in science, engineering and technology. Singapore: Springer, 2017:17－26.

[14] 郭顺利,张向先. 面向中文图书评论的情感词典构建方法研究[J]. 现代图书情报技术,2016,32(2):67－74.

[15] MOGHADDAM S, ESTER M. Opinion digger: an unsupervised opinion miner from unstructured product reviews[C]// ACM international conference on information and knowledge management,

New York:ACM, 2010:1825 –1828.
[16] 刘丽珍,赵新蕾,王函石,等. 基于产品特征的领域情感本体构建[J]. 北京理工大学学报,2015,35(5):538 –544.
[17] WANG H, NIE X, LIU L, et al. A fuzzy domain sentiment ontology based opinion mining approach for Chinese online product reviews[J]. Journal of computers, 2013, 8(9):2225 –2231.
[18] ZHU J, WANG H, ZHU M, et al. Aspect-based opinion polling from customer reviews[J]. IEEE transactions on affective computing, 2011, 2(1):37 –49.
[19] YAN Z, XING M, ZHANG D, et al. EXPRS: an extended pagerank method for product feature extraction from online consumer reviews[J]. Information & management, 2015, 52(7):850 –858.
[20] PANG B, LEE L,VAITHYANATHAN S. Thumbs up?: sentiment classification using machine learning techniques[C]//Proceedings of the ACL –02 conference on empirical methods in natural language processing-volume 10. Stroudsburg: Association for Computational Linguistics, 2002: 79 –86.
[21] DAVIDOV D, TSUR O, RAPPOPORT A. Enhanced sentiment learning using Twitter hashtags and smileys[C]// International conference on computational linguistics: posters. Stroudsburg: Association for Computational Linguistics, 2010:241 –249.
[22] 苏莹,张勇,胡珀,等. 基于朴素贝叶斯与潜在狄利克雷分布相结合的情感分析[J]. 计算机应用,2016,36(6):1613 –1618.
[23] WANG S, MANNING C D. Baselines and bigrams: simple, good sentiment and topic classification[C]// Meeting of the Association for Computational Linguistics: short papers. Stroudsburg: Association for Computational Linguistics, 2012:90 –94.
[24] 陈钊,徐睿峰,桂林,等. 结合卷积神经网络和词语情感序列特征的中文情感分析[J]. 中文信息学报,2015,29(6):172 –178.
[25] FAN L, ZHANG Y, DANG Y, et al. Analyzing sentiments in Web 2.0 social media data in Chinese: experiments on business and marketing related Chinese Web forums[J]. Information technology & management, 2013, 14(3):231 –242.
[26] 何跃,朱婷婷. 基于微博情感分析和社会网络分析的雾霾舆情研究[J]. 情报科学,2018(7):91 –97.
[27] 安璐,吴林. 融合主题与情感特征的突发事件微博舆情演化分析[J]. 图书情报工作,2017,61(15):120 –129.
[28] 由丽萍,王嘉敏. 基于情感分析和VIKOR多属性决策法的电子商务顾客满意感测度[J]. 情报学报,2015,34(10):1098 –1110.
[29] 首欢容,邓淑卿,徐健. 基于情感分析的网络谣言识别方法[J]. 数据分析与知识发现,2017(7):44 –51.
[30] 肖璐, 陈果, 刘继云. 基于情感分析的企业产品级竞争对手识别研究——以用户评论为数据源[J]. 图书情报工作, 2016, 60(1):83 –90.
[31] 朱继召, 贾岩涛, 徐君, 等. SparkCRF:一种基于Spark的并行CRFs算法实现[J]. 计算机研究与发展, 2016, 53(8):1819 –1828.
[32] CHEN J, LI K, TANG Z, et al. A parallel random forest algorithm for big data in a Spark cloud computing environment[J]. IEEE transactions on parallel & distributed systems, 2017, 28(4):919 –933.
[33] HAI M, ZHANG Y, ZHANG Y. A performance evaluation of classification algorithms for big data[J]. Procedia computer science, 2017,122(1):1100 –1107.
[34] 宋杰, 孙宗哲, 毛克明, 等. MapReduce大数据处理平台与算法研究进展[J]. 软件学报, 2017, 28(3):514 –543.
[35] SALLOUM S, DAUTOV R, CHEN X, et al. Big data analytics on Apache Spark[J]. International journal of data science & analytics, 2016, 1(3/4):145 –164.
[36] 邢晓宇. 决策树分类算法的并行化研究及其应用[D]. 昆明:云南财经大学, 2010.
[37] 卫洁. MapReduce框架下的贝叶斯文本分类学习研究[D]. 太原:山西财经大学, 2012.
[38] 罗元帅. 基于随机森林和Spark的并行文本分类算法研究[D]. 成都:西南交通大学, 2016.
[39] 刘泽燊, 潘志松. 基于Spark的并行SVM算法研究[J]. 计算机科学, 2016, 43(5):238 –242.
[40] THINKNOOK. Twitter sentiment analysis training corpus (dataset)[EB/OL]. [2017 –12 –30]. http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/.
[41] GO A, BHAYANI R, HUANG L, Sentiment140[EB/OL]. [2017 –12 –30]. https://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip.
[42] HEREDIA B, KHOSHGOFTAAR T M, PRUSA J, et al. Cross-domain sentiment analysis: an empirical investigation[C]//International conference on information reuse and integration. New York: IEEE, 2016:160 –165.
[43] GOEL A, GAUTAM J, KUMAR S. Real time sentiment analysis of Tweets using Naive Bayes[C]// International conference on next generation computing technologies. New York: IEEE, 2016:257 –261.
[44] LIMA M L, NASCIMENTO T P, LABIDI S, et al. Using sentiment analysis for stock exchange prediction[J]. International journal of artificial intelligence & applications, 2016, 7(1):59 –67.
[45] FRIEDRICH N, BOWMAN T D, STOCK W G, et al. Adapting sentiment analysis for tweets linking to scientific papers[EB/OL]. [2017 –12 –30]. http://cn.arxiv.org/pdf/1507.01967v1.

作者贡献说明:

余传明:论文构思、数据获取、论文初稿撰写、论文修改;
原赛:机器学习对比实验、论文初稿撰写、论文修改;
王峰:大数据平台构建、论文修改;
安璐:论文构思、论文修改。

余传明，原赛，王峰，等.大数据环境下文本情感分析算法的规模适配研究：以Twitter为数据源[J].图书情报工作,2019,63(4):101－111.

Research on Scale Adaptation of Text Sentiment Analysis Algorithm in Big Data Environment: Using Twitter as Data Source

Yu Chuanming[1] Yuan Sai[2] Wang Feng[1] An Lu[3]

[1] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073

[2] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan 430073

[3] School of Information Management, Wuhan University, Wuhan 430072

Abstract: [**Purpose/significance**] This paper aims to study the scale adaptation problem for the purpose of textual sentiment analysis in big data environment. The paper provides reference for the best choice between efficiency and cost when researchers in the field of information science carry out data analysis under big data environment. [**Method/process**] We use the Sentiment140 dataset of Stanford University. Based on the analysis of traditional sentiment analysis algorithms, we propose five textual sentiment analysis algorithms for big data to test the adaptation effectiveness of various algorithms under different environments and data sizes, and conduct empirical comparisons in terms of accuracy, scalability and efficiency. [**Result/conclusion**] The experimental results show that the cluster built in this paper has good operational efficiency, correctness, and scalability. Spark clusters have more efficiency advantages in processing large-scale text sentiment analysis data, and with increasing the data size, its efficiency advantage is more obvious. In resource utilization, as the number of nodes and cores increase, the overall operating efficiency of the cluster changes significantly. We find the configuration of five slave nodes with 4 cores and 4G memory can achieve the effect of saving resource costs while efficiently completing the classification task.

Keywords: scale adaptation big data massive text sentiment analysis machine learning algorithm

关于在学术论文署名中常见问题或错误的诚信提醒

恪守科研道德是从事科技工作的基本准则,是履行党和人民所赋予的科技创新使命的基本要求。中国科学院科研道德委员会办公室根据日常科研不端行为举报中发现的突出问题,总结当前学术论文署名中的常见问题和错误,予以提醒,倡导在科研实践中的诚实守信行为,努力营造良好的科研生态。

提醒一:论文署名不完整或者夹带署名。应遵循学术惯例和期刊要求,坚持对参与科研实践过程并做出实质性贡献的学者进行署名,反对进行荣誉性、馈赠性和利益交换性署名。

提醒二:论文署名排序不当。按照学术发表惯例或期刊要求,体现作者对论文贡献程度,由论文作者共同确定署名顺序。反对在同行评议后、论文发表前,任意修改署名顺序。部分学科领域不采取以贡献度确定署名排序的,从其规定。

提醒三:第一作者或通讯作者数量过多。应依据作者的实质性贡献进行署名,避免第一作者或通讯作者数量过多,在同行中产生歧义。

提醒四:冒用作者署名。在学者不知情的情况下,冒用其姓名作为署名作者。论文发表前应让每一位作者知情同意,每一位作者应对论文发表具有知情权,并认可论文的基本学术观点。

提醒五:未利用标注等手段,声明应该公开的相关利益冲突问题。应根据国际惯例和相关标准,提供利益冲突的公开声明。如资金资助来源和研究内容是否存在利益关联等。

提醒六:未充分使用志(致)谢方式表现其他参与科研工作人员的贡献,造成知识产权纠纷和科研道德纠纷。

提醒七:未正确署名所属机构。作者机构的署名应为论文工作主要完成机构的名称,反对因作者所属机构变化,而不恰当地使用变更后的机构名称。

提醒八:作者不使用其所属单位的联系方式作为自己的联系方式。不建议使用公众邮箱等社会通讯方式作为作者的联系方式。

提醒九:未引用重要文献。作者应全面系统了解本科研工作的前人工作基础和直接相关的重要文献,并确信对本领域代表性文献没有遗漏。

提醒十:在论文发表后,如果发现文章的缺陷或相关研究过程中有违背科研规范的行为,作者应主动声明更正或要求撤回稿件。

院属各单位应根据以上提醒,结合本单位学科特点和学术惯例,对科研人员进行必要的教育培训,让每一位科研工作者对学术论文署名保持高度的责任心,珍惜学术荣誉、抵制学术不端行为,将科研诚信贯穿于学术生涯始终。

来源:中国科学院监督与审计局